Escaping the design traps of creeping featurism: Introducing a fitness management strategy

Dong-Seok Lee, David D. Woods, Daniel Kidwell
Cognitive Systems Engineering Lab., The Ohio State University
210 Baker Systems, 1971 Neil Ave. Columbus OH 43210
dongseok.lee@gmail.com, woods.2@osu.edu, kidwell.18@osu.edu, 614-292-1296

ABSTRACT

Creeping featurism is a process that produces complexity. This paper describes ways to prevent or escape from creeping featurism. We discuss limitations of current methods of user interface design and usability engineering, and propose a fitness management strategy to detect feature creep and to aid the search for innovations.

INTRODUCTION

User-centered design is now widely accepted, and many companies have organized teams for usability engineering. Such teams have been involved in the product development process, and have contributed to a better usability product. Despite active usability engineering groups, some products have become complicated and difficult to use [1, 2, 3]. One example is the cell phones we use every day. At first, they were not very difficult to use, since they resembled the conventional telephone, but after converging the functions of other devices such as cameras, games, and MP3 players, users could no longer perform the tasks they wanted to; and sometimes, the devices performed tasks that the users didn't want them to.

One of the main reasons for complicated products is *creeping featurism*. Creeping featurism describes "a systematic tendency to load more chrome and features onto systems at the expense of whatever elegance they may have possessed when originally designed" [4]. Every feature or function is useful to at least someone as designers or marketers intended, but the increased number of features causes general complexity problems. For example, the number of display texts of Nokia cell phones, which reflects the number of software features, was increased to almost eight times what it was in 1992 (406 texts) to 3085 texts in 2002 [5].

The complexity-producing process of creeping featurism plagues users and designers. What is the users' reaction to complicated products? Users simplify the product; they throw away some unused features, build their own maps to avoid getting lost, and eventually find that the product was bloated to begin with. McGrenere [6] reported that 15.8% of 265 first-level functions of Microsoft MS Word were not used at all and 21.5% were used by more than half of the users. A survey by Philips [7] reported that 65 percent of people in the USA do not buy technology products because they perceive them to be too complex to set up and operate, and half of the respondents said that the electronics industry introduces whatever it thinks will sell.

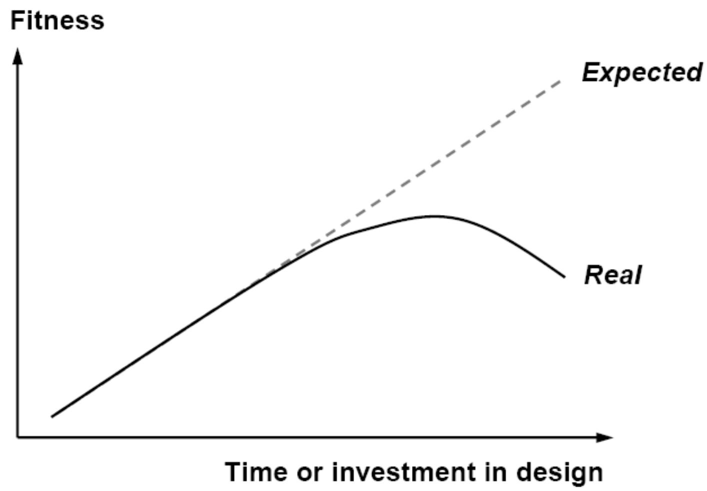
Blame from users and managers for user complaints, high return rate, and high service costs is often assigned to the designers and usability engineers of a given product. The New York Times [8] complained that a recently released Samsung cell phone with new fancy features didn't work reliably and was too difficult to use because of its complicated user interface architecture: 583 menu commands. Whose fault is this? Is this the designers' problem? Is this the problem of usability engineering? Does the problem belong to the user interface managers? Is it due to a faulty development process? Or is this problem unavoidable in a competitive market? In fact, designers and usability engineers have done their work hard, and didn't mean to make their product complex through feature creep. Then what is the problem? The authors feel that there are some traps or limitations that we have not yet considered. This paper discusses how to escape these traps.

MORE FEATURES, BETTER PRODUCTS?

The Wall street Journal [9] reported that "Mercedes acknowledges removing hundreds of electronic functions from its cars." In 2005, Vodafone introduced 'Vodafone Simply,' which provides mobile phones that have a smaller set of features: those that are used most frequently [10]. These measures look quite strange because other companies have always added new features and have advertised them. What motivates companies such as Mercedes and Vodafone to move in that direction? The authors think that Mercedes has noticed that many of their features are annoying or are ignored by their customers, and Vodafone has seen users' need for a simpler phone after being frustrated by a lot of unnecessary features. So these companies decided to invest money to remove some of the very features that they had invested money to develop.

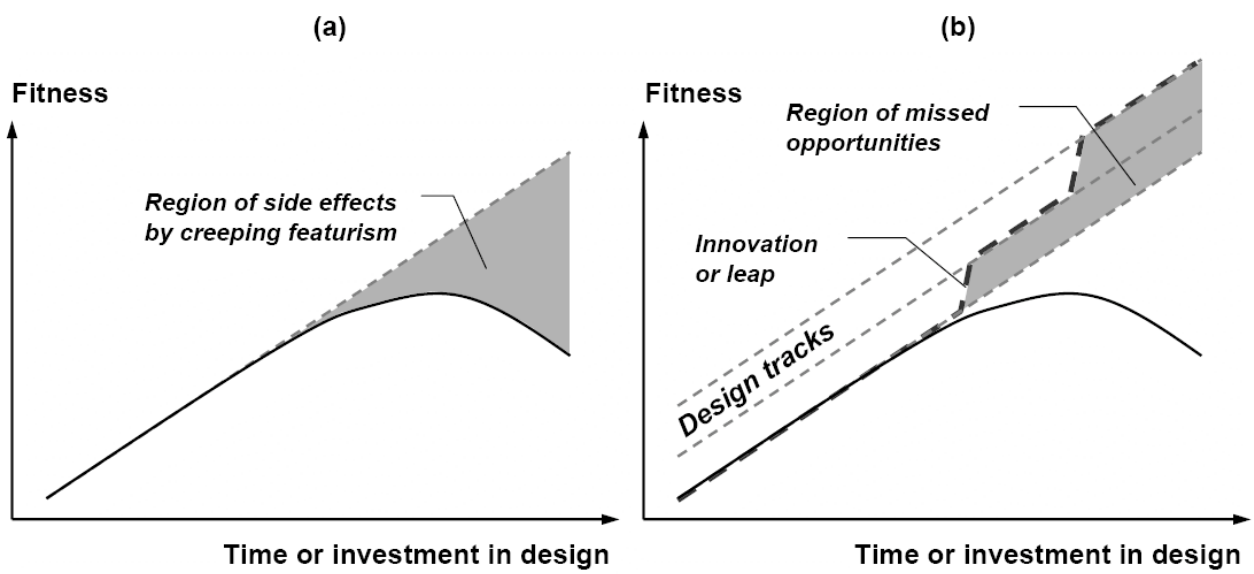
The increase in the number of features would be the natural result of an evolving process. Many people believe that "more features" means "better products," in which better implies more useful, usable, and marketable. This is true until a product reaches the point of feature creep. The *fitness* of the product increases when we invest in design efforts, but at a certain point, the fitness suddenly stops increasing and starts to decrease (Figure 1). In other words, designers expect a fitness increase by adding new features, but the complexity caused by feature creep *reduces* the fitness. Fitness refers to an individual's ability to propagate its genes in biology, or to an individual's physical condition. In this study, fitness is introduced as a measure of complexity of products. The larger fitness is, the better or more suitable is the product to the users in context. It means more than usability and is related to the match between the design and the user goals and activities. We think Mercedes and Vodafone have noticed that the fitness of their products was decreasing significantly due to the overload of features as in right-hand side of Figure 1.

Figure 1. Fitness Curve



What designers or usability engineers need is the ability to keep their eyes on the fitness of products in order to prevent or escape feature creep. They should know whether fitness is increasing or decreasing, and should notice symptoms of eventual fitness saturation and fitness decrease. The figures below provide two perspectives to be considered: (1) how can we know when complexity has begun to dominate users' reactions? and (2) how can we find new promising directions for innovation activities? The following sections will discuss these questions.

Figure 2. The side effects of feature creep (a) and missed opportunities for innovation (b)



TRACKING FITNESS FOR DETECTING FEATURE CREEP

The difference between expected fitness and real fitness reflects the side effects of feature creep (Figure 2-a). The larger the difference is, the greater the number of user complaints, the higher the service cost, the higher the return rate, and the lower the market competitiveness. How can we be aware of when complexity starts to dominate? Can usability testing, user surveys, or focus group interviews find that? The authors think the methods have limitations and do not consider all of the aspects of feature creep. Based on our findings and on recent studies about user experience, here are several suggestions:

- *Have larger- and longer-viewpoints*: When companies call outside organization for usability engineering focuses on one-time improvement. Usability engineering inside companies should have wider points of view in order to manage the evolution process of their products rather than trapped in short-term schedule pressures.
- *See how users are coping with complexity*: Not many studies investigate how users deal the complexity of user interfaces. Rather than just measuring task performance data, it is crucial to understand what difficulty users confront while interacting with products and what strategies they deploy.
- *Assess what will be made worse by adding new features, don't just assess what will be made better*: Evaluation usually focuses on the bright side of changes, but there may be a dark side.
- *Think about what to remove as well as what to add*: In many product development organizations, everyone looks for new features to add, but no one considers what to remove. It is essential that development teams check what features are not used or are used infrequently.
- *Assess the reasons for new usability problems or problems that have become more serious*: New usability problems or problems that have become more serious could be symptoms of feature creep.
- *Focus on the value of products rather than on new features*: What is important is the overall value of products for users, not their new features. Experimental settings or survey sheets are often designed to show the merits of new features.
- *Analyze the contents of user complaints from web sites, sales people, or service calls*: When complexity dominates, users look for simpler products or alternatives.
- *Manage the history of user interface changes and the reasons for the changes*: Feature creep is a process of making complexity over a series of design decisions. It is necessary to know the detailed story of design changes, intent, and impact in order to detect and respond to complexity.

KLONDIKE SEARCH FOR INNOVATION

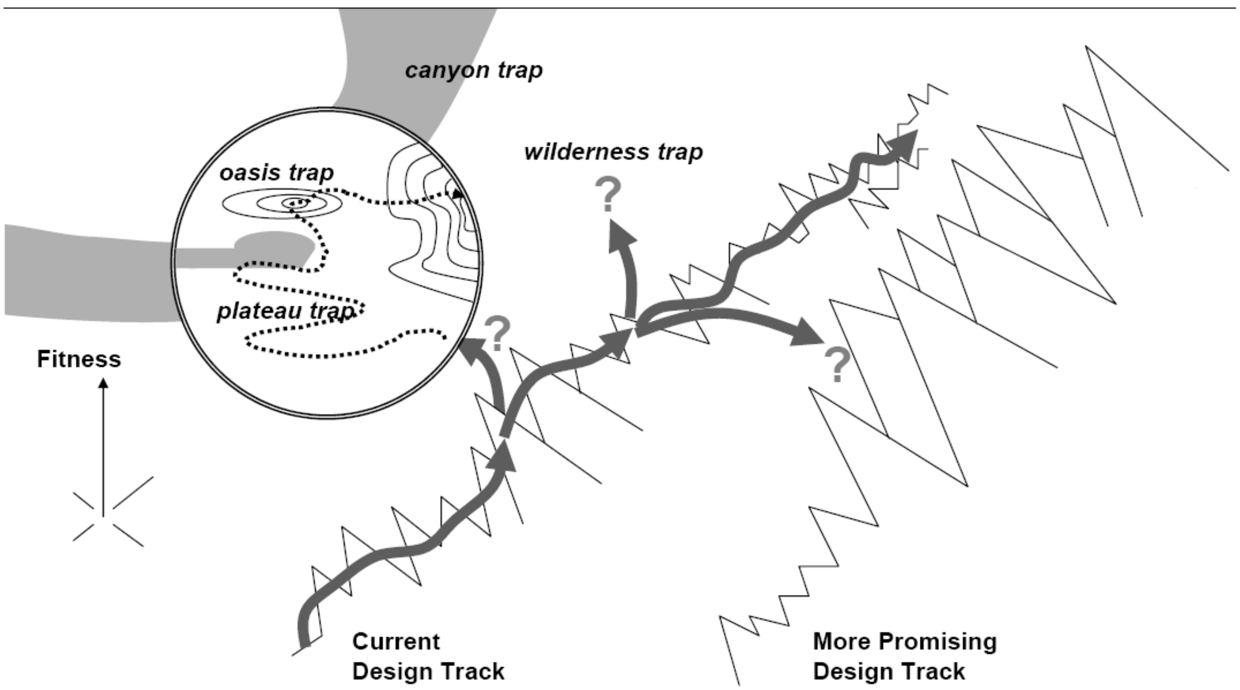
We all look for a better user interface or innovation. Another perspective proposed in this study is the concept of *design track*. Because of a high competitive market, fast-paced development, the large number of developing products, and small changes in newer versions or generations, practitioners do not have enough time or budget to follow a full user-centered process. Thus, designers usually added menu items or buttons in order to add new features to ancestors, and it was difficult for usability engineers to change the design even if they found serious usability problems. Design track refers to the idea that once a product is designed, it frames and limits future design decision. Thus the range of possible successor products can become narrowed around the dimensions and structure of the previous design. Designer's becomes "stuck" in this frame or channel creating a design track. As the track matures (Figure 2-b), design changes produce complexity and less value for users (fitness decrease). For example, clear distinction between functions of the 2003 model of Samsung's multi-function printer (Figure 3-a) was broken when the 2004 model, its successor, added many new features (Figure 3-b).

Figure 3. Design changes of a multi-function printer

Moving to a new design track is a conceptual shift. Design activity moves in new directions that appear promising as shown in Figure 2-b. For example, while other MP3 player manufacturers were trapped in the design of buttons and joysticks, Apple introduced the 'Wheel' and 'Click Wheel' interface to iPod, its MP3 player, which provides users with great convenience for manipulating a large number of music files. Apple iPod has converged new features such as picture view, text reading, and game, but all could be integrated around the circular interface concept. It's a sign of no feature creep that users added new features themselves around the basic capabilities and interface concepts. This means that the design track of iPod has more fitness and more endurance for feature creep. The danger is that pressures to improve while working on one design track may prevent designers from searching for or even seeing other potentially promising design tracks that could lead to innovations and jumping to new fitness curves (Figure 2-b)

Figure 4 graphically explains the idea of design tracks in a *fitness landscape*. Current designs follow the design track on the left, feature creep was detected so that fitness is started to decrease, but there is a better track in the middle. The fitness landscape is taken in part from evolutionary theory to contemporary cognitive science and complexity theory. The idea is that the design track contains all of the possible forms of a technology, represented as points in an abstract space or landscape. While the landscape has many dimensions, it can be thought of in terms of a three-dimensional topographic map. Within the landscape, various design ideas, capabilities, features and forms take on a level of fitness – some more "fit" than others, depending on the criteria. The degree of fitness of a particular design track is represented by a height above the corresponding point on the landscape. We can thus think of the landscape where a high mountain peak area would stand for a particularly fit idea, capability, feature and/or form [11]. Finding what is most promising with the design track becomes the process of traversing the landscape of ideas, capabilities, features and forms in search of such high point peaks. By having a more fit design track, we can move ideas, capabilities, features and forms from regions of missed opportunities, design traps and creeping featurism to leaps onto higher regions, innovations and explorations within the fitness landscape.

Figure 4. Design tracks with a Klondike search on a fitness landscape.



Once complexity begins to dominate, design problems cannot be solved by iteration of refinements within a design track. At this point, additional iterative refinement is likely only to increase complexity for users and decrease fitness (Figure 1). Iteration of fixing problems after usability testing also does not guarantee better products, since feature creep will generate more and serious problems. Ironically usability resolve individual issues associated with one or a few features, but evidence of decreasing fitness grows in the form of user complaints, return rate, and service costs. Further investment in usability seems ineffective or even counter productive leading company management to question the value of their usability groups. Roesler at al. [12] refer to usability engineering as following standard prescriptive approaches to technology development by solving short-term small repair problems at a local scale, and usually having to swim upstream relative to pressure on design, thus being trapped in a narrowing solution space.

Escaping feature creep, then, requires determining how to discover other design tracks, how to assess the "promisingness" of tracks or regions, and deciding when to leap to promising tracks. Perkins [11] provides a good model for these questions: *Klondike search* (Figure 4). He classifies design into homing search (new features and iterative refinement still produce increases in fitness) and Klondike search (fitness has saturated and improvements require a shift to a new design track). The homing search can find solutions by following pointing directions from contours and gradients like incremental refinements. However incremental refinements don't guarantee finding solutions in Klondike space, since the space has few solutions (wilderness trap); many regions have no clues as to direction (plateau trap); there are barriers such as cost, which isolate the areas to be searched (canyon trap); and there are regions that have better local fitness (oasis trap). Design space is a kind of Klondike space; design for innovation, then, requires different search strategies. Below are some suggestions based on the characteristics of Klondike search.

- *Pre-explore design space*: Finding design solutions in Klondike space requires a long search, since design space is large, with few solutions. Thus, after feature creep takes place, it is too late to find a better design. In many cases, insights are preceded by long searches. Pre-explore design space as much as possible, assess the promisingness of the explored region, and build maps.
- *Explore randomly*: After a series of incremental refinements, the design may be located on local optimum and surrounded by constraints. To avoid being trapped in local optimums, it is essential that we ask unreasonable questions and break constraints.
- *Be patient*: Breakthrough arrives suddenly with little or no apparent progress, since many design states have no clues pointing in a direction. Progress only occurs when we approach a solution.
- *Check environmental changes from technology and users*: Technology changes may break barriers to solutions. Also, people change over time, and this changes design space.
- *Continually reassess promisingness of under-explored areas*: Environment around the product itself changes over time as Berger and Heath noticed [13], and changes fitness landscape and fitness of products.

FITNESS MANAGEMENT

The unifying theme of this paper is *fitness management* and how it can help practitioners escape the design complexities of creeping featurism. Fitness management is where a product's design track is explored, new possibilities are envisioned, and promising directions are scouted as described earlier with the Klondike search. Table 1 lists a set of comparative operating terms used in fitness management; they are organized to contrast usability engineering versus fitness management. One example in current usability engineering practice is to use a static approach to solving short-term repair problems with one iteration, one at a time, at a local scale. In contrast, fitness management has a broadening point-of-view at a regional level of scale with a dynamic problem-solving slant. A dynamic approach would be to have multiple shifting approaches that converge, diverge, and rove, detecting and reframing ideas relative to an ongoing debate about what is "promising."

The list of terms on the left side are properties of usability engineering which lead it to be unable to recognize and redesign ideas, capabilities, features and forms in order to reduce or avoid complexity. Notice that some of the properties of usability engineering describe the dimensions of what usability engineers struggle with – having to solve short-term problems and repair issues. Fitness management is not about the repair of concepts or problems. It is, more importantly, broadening idea finding beyond the current activity set or problem-solving track. Fitness management has the ability to make a broader diagnosis of what is taking place, e.g. recognizing the signals of saturation or notifying an imbalance between homing and Klondike search. Therefore fitness management has the ability to see new directions, and discover new opportunities.

Table 1. Properties of Usability Engineering through Fitness Management

Usability Engineering	Fitness Management
Static – Problem-solving with single iteration of one idea. Solving one problem at-a-time with limited viewpoint.	**Dynamic –** Having multiple shifting approaches that converge, diverge, roving with detecting and reframing.
Objects of change & single repair	**Vectors of change & what's promising**
Product & use-centered measure	**Relation & human-centered measure**
Narrow point-of-view (local)	**Wider point-of-view (regional)**
Problem repair	**Idea finding & discovering opportunities**
Inherent complexity	**Managing complexity**
Homing search in problem space	**Klondike search in problem space**
Post-assessment	**Pre-assessment**
Brittle management	**Resilience management**
Short-term risk management	**Long-term sustainability**
Reactive	**Proactive & anticipatory**
Follow-up development	**Innovation**
Small problems	**Large problems**
Defensive management	**Aggressive offensive management**

Properties of Usability Engineering and Fitness Management

CONCLUSION

Creeping featurism is a critical problem that products face. What is the opposite of creeping featurism, then? This paper describes the authors' journey toward finding an answer to that question. Creeping featurism, again, is the process of producing complexity overall despite a series of local improvements. This study focuses on how usability engineers and designers are trapped in process of feature creep. The real pressures on commercial development usually leave usability groups trapped in local incremental improvement process: designers add new features to existing designs, and usability engineers test and fix usability problems through evaluation. It is asserted with concepts and examples that their strategies of incremental refinements can neither prevent nor help to escape feature creep.

First, this study introduced fitness as a measure of the complexity of products. When indicators of fitness start to level off or decrease, this probably reveals fitness saturation. Tracking changes in fitness is imperative, and this requires a shift in perspective that is outside the capacity of current usability methods. Several suggestions for tracking fitness were proposed. Secondly, this study has shown the need to shift to new design tracks when there is evidence of feature saturation. Characteristics of Perkins' Klondike search were used to suggest possible strategies. Finally, the philosophy of fitness management was presented and compared to that of usability engineering. Fitness management complements usability engineering by providing longer term and broader view points for tracking changes in product fitness and for assessing the investment balance between homing search and Klondike search. The key challenge for fitness management is to develop participatory methods for assessing "promisingness" of new directions.

Further study will explore managerial and organizational issues about fitness management. Adding fitness management as an innovation strategy to your process is a way to avoid feature creep and inadvertent complexities with your products. An organizational strategy is to introduce fitness management as an organizational unit, program, or function within your usability group. It can act as a bridge group between research and product development, as part of a matrix organization where it becomes a coordination organization with R&D, usability engineering, user interface, user experience, marketing, and design. Fitness management is a new way for an organization to look at its strategic and tactical functions.

REFERENCES

1. Cooper, A. (1999). *The Inmates are Running the Asylum: Why High Tech Products Drive Us Crazy and How to Restore the Sanity*. Indiana: Sams.

2. Nielsen, J. (2004). Why Consumer Products Have Inferior User Experience. *Alertbox, March 15, 2004*. Available from: http://www.useit.com/alertbox/20040315.html.

3. Bogaards, P. (2005). User Experience: Back to Business. *Interactions, 12 (3),* pp. 23-25.

4. FOLDOC. Available from: http://foldoc.doc.ic.ac.uk/foldoc/foldoc.cgi?creeping+featurism.

5. Lindholm, C., Keinonen, T. and Kiliander, H. (2003). *Mobile Usability: How Nokia Changed the Face of the Mobile Phone*, NY: McGraw-Hill, pp. 12.

6. McGrenere, J. (2002). *The Design and Evaluation of Multiple Interfaces: A Solution for Complex Software*. Unpublished Ph.D. Dissertation. Department of Computer Science, University of Toronto, Toronto, ON.

7. Philips. (2004). Consumers Say Technology Development Should Center Around Their Needs. *Press Information, Nov 10, 2004*, pp. 2. Available online at http://www.futureguru.com/docs/philipspr.pdf

8. Pogue, D. (2005). The Cellphone That Does Everything Imaginable, at Least Sort Of. *NY Times May 12, 2005*.

9. Power, S. (2005). There' s No Place Like...Cars? *Wall Street Journal, July 25, 2005*.

10. Vodafone (2005). *Introducing Vodafone Simply.* Available from: http://shop.vodafone.co.uk/simply.

11. Perkins, D. (2000). *The Eureka Effect: The Art and Logic of Breakthrough Thinking*. NY: W. W. Norton and Company. pp. 67-87.

12. Roesler, A., Woods, D., and Feil, M. (2005). Inventing the Future of Cognitive Work: Navigating the 'Northwest Passgage' of Design. In W. Jonas, R. Chow, N. Verhaag (eds.) *Design-System-Evolution: Application of systemic and evolutionary approaches to design theory, design practice, design research and design education*. European Academy of Design, Bremen, Germany.

13. Berger, J. and Heath, C. (2005). Idea Habitats: How the Prevalence of Environmental Cues Influences the Success of Ideas. *Cognitive Science*, Vol. 29, No. 2, pp. 195-221.

Acknowledgements

Prepared, in part, through collaborative participation in the Advanced Decision Architectures Consortium sponsored by the U. S. Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-2-0009. This research is also supported in part by the Ministry of Information and Communication of the Republic of Korea.